(1)

SOFTWARE COST ESTIMATING
MODELS: A CALIBRATION, VALIDATION,
AND COMPARISON

THESIS

Gerald L. Ourada, Captain, USAF

AFIT/GSS/LSY/91D-11

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSS/LSY/91D-11

SOFTWARE COST ESTIMATING MODELS: A CALIBRATION, VALIDATION, AND COMPARISON

THESIS

Gerald L. Ourada, Captain, USAF

AFIT/GSS/LSY/91D-11

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GSS/LSY/91D-11

SOFTWARE COST ESTIMATING MODELS: A CALIBRATION, VALIDATION, AND COMPARISON

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Software Systems Management

Gerald L. Ourada, B.S.

Captain, USAF

December 1991

Preface

This thesis effort was an analysis of four software effort estimation models. I performed a calibration and validation of the models in one development environment and then a comparison using another development environment. I hoped to show that several of the models we currently use at the program office level are fairly good estimators of software development projects. I found this not to be the case. I found the models to be highly inaccurate and very much dependent upon the interpretation of the input parameters.

I originally started this effort to educate myself on the various models and their application to a Air Force System Program Office. I no longer have faith in the estimates the "experts" have been giving me for the last 10 years.

I am deeply indebted to my thesis advisor, Mr. Dan Ferens, for his help, guidance, and encouragement. I also owe a big "thanks" to Capt. Robbie Martin (SSD/ACC) for providing me a credible database to work with. And even though it arrived too late to use in this effort, a big thanks to Ms. Gayla Walden (Aerospace Corp.) for getting the Aerospace software histories database to me.

And last but definitely not least, to my wife, thanks SWEETHEART for the support over the last 18 months. I couldn't have done it without you.

Gerald L. Ourada

Table of Contents

List of Tables

Abstract

This study was a calibration, validation and comparison of four software effort estimation models. The four models evaluated were REVIC, SASET, SEER, and COSTMODL. A historical database was obtained from Space Systems Division, in Los Angeles, and used as the input data. Two software environments were selected, one used to calibrate and validate the models, and the other to show the performance of the models outside their environment of calibration.

REVIC and COSTMODL are COCOMO derivatives and were calibrated using Dr. Boehm's procedure. SASET and SEER were found to be uncalibratable for this effort. Accuracy of all the models was significantly low; none of the models performed as expected. REVIC and COSTMODL actually performed better against the comparison data than the data from the calibration. SASET and SEER were very inconsistent across both environments.

SOFTWARE COST ESTIMATING
MODELS: A CALIBRATION, VALIDATION,
AND COMPARISON

I. Introduction

Overview

With the tremendous growth of computers and computer software over the last 20 years, the ability to predict the cost of a software project is very critical to management both within the Department of Defense (DoD) and the civilian industry. In 1980, approximately $40 billion, or 2 percent of the Gross National Product, was spent on software products (3:1462). "With estimates of 12% per year growth, the 1990 expenditures on software will be $125 billion nationwide" (3:1462). The DoD expected to purchase as much as $30 billion of software products in 1990 (9:15). Managers with this amount of money tied up in software procurement must be able to predict how much a particular software project will cost. In the military, "Whether potential enemies are deterred or battles are won or lost will depend increasingly in the future on complex computer software" (9:15). This was clearly evident in the recent Desert Shield/Storm war. In the Aviation Week and Space

Technology summary articles of the war, four keys to the success of the air power were identified:

1. Highly accurate navigation and weapon delivery systems;
2. Stealth technology, embodied in the F-117;
3. Night attack systems to maintain pressure around-the-clock;
4. Surveillance and intelligence-gathering systems, such as AWACS, Joint-STARS, space systems and tactical reconnaissance aircraft (20:42).

All of the above mentioned systems are highly dependent upon software for their functionality. What better reason do we, as military leaders and procurement specialists, need to understand the issues of software procurement?

This chapter presents the research to be completed in this thesis. First, the general issue of software effort estimation will be covered; second, the specific issue and research questions will be covered; and third, a discussion on the limiting scope of the research will be addressed.

General Issue

One of the biggest issues in software procurement is accurate estimation of the cost of a particular software project. Cost estimates must be used in two key areas. The first area covers costs e imated during project conception. These estimates are used for budgetary purposes, i.e. submissions to Congress, and to compare

against proposal submissions. Second, are those estimates used throughout the project life-cycle that must be continually reevaluated to accurately track on-going contracts for cost accounting purposes and to estimate completion costs. The key is to be able to accurately estimate the cost of completion of projects at any point in the life-cycle. This thesis addresses whether DoD has the necessary tools to accurately estimate analysis, design and coding, and modification of software projects.

Specific Issue

Specifically, this research effort analyzes existing software effort estimation models. Many models are used throughout the DoD, but their accuracy and usability are still questionable. These models have yet to receive a rigorous calibration and testing from a solid historical database (8:559). They also have not been used throughout a program acquisition with the necessary data collection and model analysis to show model accuracy. This research ascertains whether these models can be calibrated and validated to establish their relative accuracy.

Most models will also perform a schedule estimation along with the effort estimation. This research effort does not address the schedule estimation. (For an example of schedule estimation research see the thesis effort of Capt. Bryan Daly, "A Comparison of Software Schedule Estimators," AFIT/GCA/LSQ/90S-1, published in September 1990.)

Research Objectives

This research addresses the following set of questions:

1. Given a credible set of actual DoD data, can the chosen models be calibrated?

2. Given a calibrated model, with another set of actual data from the same environment, can the models be validated?

3. Given a validated model, if another independent data set from another software environment is used, are the estimates still accurate?

4. Is a calibration and validation of a model accurate for only specific areas of application?

Scope of Research

Since effort estimation models can be expensive, this research was limited to models existing at AFIT or available from other government sources. Currently there are eight such models

1. REVIC (REVised version of Intermediate COCOMO);
2. COCOMO (COnstructive COst MOdel);
3. PRICE-S (Programmed Review of Information for Costing and Evaluation Software);
4. SEER (System Evaluation and Estimation of Resources);
5. SASET (Software Architecture, Sizing and Estimating Tool);
6. System-4;
7. Checkpoint/SPQR-20;
8. COSTMODL (COST MODeL).

Time constraints restricted this research to four models. The following are the four selection criteria used to guide the selection of models to study:

1. Use within DoD or NASA;
2. Ease of understanding and analyzing the input and the output;
3. Availability of model documentation;
4. Cost to use the models for this research effort.

The above criteria were derived from personal experience in project management within DoD and the potential for cost to impact the research effort. Only those models that are relatively easy to use and understand will be used by any project team. Also if the model already belongs to the government, then there exists a greater chance of the model being used due to less cost to the potential user.

The four models selected were, REVIC, SASET, SEER, and COSTMODL. For each of these models, either DoD or NASA has a license to use or is the owner of the model. (SEER is to be site-licensed to the Air Force in October 1991.)

Definition of Terms

1. Calibration - The adjustment of selected parameters of a given model to get an expected output with known inputs. In the world of statistics this effort is known as model building. For this research effort, the models already exist and will only be modified.

2. Validation - Testing a specific model using known inputs and establishing the output to within some error range. This is independent and non-iterative with calibration. In the world of statistics, this is often called cross-validation since it will use a portion of an original data set kept out of the model building/calibration effort.

II. Literature Review

Introduction

This chapter examines recent publications in the area of software effort estimation and provides a summary of the specific models to be used during this research effort. Several key areas are highlighted: A comparison of different software procurements, the need for software effort modeling, and the parameters of good modeling techniques. A description of the COCOMO (COnstructive COst MOdel) is also given since it is a frequently used model and all others are often compared to it. Appendix A lists sources that this author found important to this effort. These documents were not used as quoted sources for this effort, but were found very useful for knowledge in this area. Any further research in this area should include them as part of the review and investigation effort.

Comparisons

To illustrate the effort involved in software procurement, Brenton Schlender in Fortune (22:100-101+), compared four very different software packages to show the amount of code, labor, and cost which are involved in a software project (see Table 2.1). Schlender quotes Frank King who said, "The labor content in large systems like

those in the space shuttle is equivalent to what it took to build the Great Pyramid" (22:101).

Table 2.1 Software Cost and Effort Comparisons

Project	Lines-of-code	Labor (man-years)	Cost ($ millions)
Lotus 1-2-3 v.3	400,000	263	22
Space Shuttle	25,600,000	22,096	1200
CitiBank AutoTeller	780,000	150	13.2
1989 Lincoln Continental	83,517	35	1.8

(22:100-101+)

The Need

Because of effort necessary to complete a software project, management must understand all the potential costs. Software effort estimation techniques are necessary to give managers the information to make cost-benefit analyses, breakeven analyses, or make-or-buy decisions (2:30). Estimates of software effort are as necessary as the estimates of hardware cost for any project. In fact, for computer based systems, the cost of the software is much more important than the cost of the hardware. According to Dr. Boehm, "The computer system, consisting of both hardware and software, bought today as purely hardware, generally costs the purchaser three times as much for the software portion as for the hardware" (2:17). No firm (public or private, non-profit or profit oriented) can

stay profitable unless it can estimate costs accurately before it begins a new project. One of the primary numbers studied at every DoD Defense Advisory Board (DAB) review is the cost estimate to complete the next phase of a system procurement. These reviews come at every major milestone and any other point that the DAB deems necessary (See AFR 57-1 for a more detailed review of the DoD Milestone Review process). The federal government now requires the use of cost estimating tools on all new military projects (17:11).

Software effort estimates are also necessary for real-time software management. Without a reasonably accurate estimate, a project manager has no firm basis from which to compare budgets and schedules; nor can he make accurate reports to management, the customer, or sales personnel (2:30). The ever increasing size and complexity of software projects makes accurate projections and understanding of the costs and schedules a management necessity (7:195).

Technique Parameters

Studies of software effort estimating have yielded a set of cost influence factors and relationships necessary to support practical effort estimation:

1. The number of source instructions or some other measure of program size;
2. The selection, motivation, and management of the people involved in the software process;

3. Product complexity, required reliability, database size, and other features which are not management controllable;

4. Productivity ranges;

5. The volatility of requirements (3:1465).

All software effort estimating techniques must take these factors and relationships into consideration, although each must receive a varying degree of emphasis. One key ingredient left out of the above listing is experience. All techniques in use today are based in some way upon experience, i.e. the use of a historical data base for calibration/validation (18:696). A historical database is mandatory if any organization is to use any of the current models effectively. Most organizations do not currently know what they have spent in the past to develop their software products (21:282). This is a problem throughout the software development industry and within DoD in particular. The necessary data to collect this information is usually some of the first to be cut from the contract in the interest of cost reduction. Because of an absence of credible data, current models have a severe deficiency in proven accuracy. Model users are lucky if they can estimate cost to within 20% of the actuals, 70% of the time (2:32; 1:1). This accuracy must increase if management is to place any confidence in the model estimates. If software can be "engineered" then any effort

estimation model should be able to predict the potential cost of a software project with a high degree of accuracy. Chapter III presents the discussion on accuracy requirements.

COCOMO Description

COCOMO, the model to which, according to Miyazaki, "all others are compared," is considered a milestone in software engineering (19:292). The input and output are much more precise and clear than many other models and techniques, and it allows for easy tailoring to the specific purpose and historical databases (19:292). COCOMO's developer, Dr. Barry Boehm, describes the model in his book Software Engineering Economics (2). He presents a hierarchy of versions: Basic COCOMO, Intermediate COCOMO, and Detailed COCOMO. Each version has three modes: organic, semi-detached, or embedded. Which mode to use is determined by the type of software being developed. The level of sophistication, flexibility, and accuracy increase as the hierarchy is climbed; but so also does the level of complexity. The Basic COCOMO model in the organic mode will be summarized here since the other versions and modes are similar to it (For further reading on any of the COCOMO models, see Dr. Boehm's book).

The Basic organic COCOMO consists of two simple effort and schedule equations.

$$MM=2.4\times(kDSI)^{1.05} \quad \text{Eq. 2.1}$$

$$TDEV=2.5\times(MM)^{0.38} \quad \text{Eq. 2.2}$$

Equation 2.1 is the basic effort equation, where KDSI is the number of thousands of delivered source instructions in the software product. MM is the number of man-months estimated for the development phase of the software life-cycle, subject to the definitions and assumptions which are described below. Equation 2.2 is the basic schedule equation, where TDEV is the number of months estimated for the software product development, subject to the same definitions and assumptions (2:61-62).

Any of the COCOMO models will provide information for any particular software project with the appropriate tailoring. The accuracy of the estimate depends upon the accuracy of the inputs, specifically the lines of code (a major point of contention for all models and languages is the exact definition of a line-of-code). One study, conducted by Miyazaki and Mori of Fujitsu Limited, has shown that with proper tailoring and use of historical databases, COCOMO can be accurate, but still does not suffice. This study showed COCOMO to predict 68% of the

database to within 20% of the actual effort value (19:299). This magnitude of error leaves a lot of room for subsequent miscalculation of the necessary resources to complete a software project. It also leaves a lot of room for improvements in software effort estimation techniques.

Analysis Model #1, REVIC

REVIC (REVised version of Intermediate COCOMO) is a direct descendent of COCOMO. There are several key differences between REVIC and the 1981 version of COCOMO, however:

1. REVIC adds an Ada development mode to the three original COCOMO modes; Organic, Semi-detached, and Embedded.
2. REVIC includes Systems Engineering as a starting phase as opposed to Preliminary Design for COCOMO.
3. REVIC includes Development, Test, and Evaluation as the ending phase, as opposed to COCOMO ending with Integration and Test.
4. The REVIC basic coefficients and exponents were derived from the analysis of a database of completed DoD projects. On the average, the estimates obtained with REVIC will be greater than the comparable estimates obtained with COCOMO.
5. REVIC uses PERT (Program Evaluation and Review Technique) statistical techniques to determine the lines-of-code input value. Low, high, and most

probable estimates for each program component are used to calculate the effective lines-of-code and the standard deviation. The effective lines-of-code and standard deviation are then used in the estimation equations rather than the linear sum of the line-of-code estimates.

6. REVIC includes more cost multipliers than COCOMO. Requirements volatility, security, management reserve, and an Ada mode are added (16:1-5).

Analysis Model #2, SASET

SASET (Software Architecture, Sizing and Estimating Tool) is a forward chaining, rule-based expert system using a hierarchically structured knowledge database of normalized parameters to provide derived software sizing values (24:1-2). These values can be presented in many formats to include functionality, optimal development schedule, and manloading charts. SASET was developed by Martin Marietta Denver Aerospace Corp. on contract to the Naval Center for Cost Analysis. To use SASET, the user must first perform a software decomposition of the system and define the functionalities associated with the given software system.

SASET uses a tiered approach for system decomposition. Tier I addresses software developmental and environmental issues. These issues include the class of the software to be developed, programming language, developmental schedule,

security, etc. Tier I output values represent preliminary budget and schedule multipliers (24:1-2 to 3-24).

Tier II specifies the functional aspects of the software system, specifically the total lines-of-code (LOC). The total LOC estimate is then translated into a preliminary budget estimate and preliminary schedule estimate. The preliminary budget and schedule estimates are derived by applying the multipliers from Tier I to the total LOC estimate (24:1-2 to 3-24).

Tier III develops the software complexity issues of the system under study. These issues include: level of system definition, system timing and criticality, documentation, etc. A complexity multiplier is then derived and used to alter the preliminary budget and schedule estimates from Tier II. The software system effort estimation is then calculated (24:1-2 to 3-24).

Tier IV and V are not necessary for an effort estimation. Tier IV addresses the in-scope maintenance associated with the project. The output of Tier IV is the monthly manloading for the maintenance life-cycle. Tier V provides the user with a capability to perform risk analysis on the sizing, schedule and budget data (24:1-2 to 3-24).

The actual mathematical expressions used in SASET are published in the User's Guide, but the Guide is very

unclear as to what they mean and how to use them (24:1-2 to 3-24).

Analysis Model #3, SEER

SEER (System Evaluation and Estimation of Resources) is a proprietary model owned by Galorath Associates, Inc. This model is based upon the initial work of Dr. Randall Jensen. The mathematical equations used in SEER are not available to the public, but the writings of Dr. Jensen make the basic equations available for review (see the two Jensen articles referenced in the bibliography).

The basic equation, Dr. Jensen calls it the "software equation" is:

$$s_e = c_{te}\sqrt{k}\,t_d \qquad 2.3$$

where s_e is the effective lines of code, c_{te} is the effective developer technology constant, k is the total life cycle cost (man-years), and t_d is the development time (years) (14:1-4). This equation relates the effective size of the system and the technology being applied by the developer to the implementation of the system (13:2-3). The technology factor is used to calibrate the model to a particular environment. This factor considers two aspects of the production technology -- technical and environmental. The technical aspects include those dealing with the basic development capability: organization

capabilities, experience of the developers, development practices and tools etc. The environmental aspects address the specific software target environment: CPU time constraints, system reliability, real-time operation, etc. (13:1-7; 23:5-1 to 5-14).

Analysis Model #4, COSTMODL

COSTMODL (COST MODeL) is a COCOMO based estimation model developed by the NASA Johnson Space Center. The program delivered on computer disk for COSTMODL includes several versions of the original COCOMO and a NASA developed estimation model KISS (Keep It Simple, Stupid) (6:2). The KISS model will not be evaluated here, but it is very simple to understand and easy to use; however, the calibration environment is unknown.

The COSTMODL model includes the basic COCOMO equations and modes, along with some modifications to include an Ada mode and other cost multipliers. The COSTMODL as delivered includes several calibrations based upon different data sets. The user can choose one of these calibrations or enter user specified values. The model also includes a capability to perform a self-calibration. The user enters the necessary information and the model will "reverse" calculate and derive the coefficient and exponent or a coefficient only for the input environment data. The model uses the COCOMO cost multipliers and does not include more as does REVIC (6:1-11).

The model includes all the phases of a software life cycle. PERT techniques are used to estimate the input lines-of-code in both the development and maintenance calculations (6:1-11).

Summary

This chapter reviewed current literature in software cost estimation. It compared different software purchases showing large differences in size and effort, reviewed the need for software cost estimation techniques, reviewed the basic parameters of all software cost estimating techniques, and summarized the models to be used in this research effort. Accurate estimates of software projects will remain a very important issue for all involved in the software engineering disciplines.

III. Methodology

Introduction

This chapter addresses the data and methodology used for the calibration, validation, and comparison of the models reviewed in Chapter 2, and the statistical tests used for accuracy analysis.

Data

The first block of historical data planned for use with this project is from Electronic Systems Division (ESD) at Hanscom AFB. This data is considered proprietary and cannot be released to non-government personnel. (For further information on this data base contact Peggy Wells, ESD/ACCT, Hanscom AFB, MA 02176.) This data base is referred to as the ESD data base throughout this thesis.

The ESD data base consists of 24 different projects, all of which were software acquisition contracted efforts managed at ESD. For each project the data base contairs the Source Lines of Code (SLOC), effort in man-months, the amount of time to complete the project, and other data necessary for analysis/use of the models. This data was considered for use for the model calibration and validation, but was eventually found to be unusable (see Chapter IV for a complete discussion on the problems with the database).

The second set of historical data was received from Space Systems Division (SSD). This data base is referred to as the SSD data base throughout this thesis. This data was to be used for comparing output of the models outside of their environment of calibration, but was eventually used for the entire effort.

Both of these databases lack some of the information for several of the model variables. Values of "nominal" were used in every model where there was no data available to make a better choice.

Methodology

This research was conducted in three parts: model calibration, validation, and comparison. During calibration the model parameters were adjusted to give an accurate output with known inputs. One-half of the database, selected at random, was used as input data. The model parameters were then adjusted mathematically to give an output as close as possible to the actual output contained in the data base. The particular calibration technique is dependent upon the particular model under evaluation; the technique suggested in the model users guide was used. Once the model was calibrated, the model was analyzed with the calibration data set to examine the model for accuracy against the calibration data.

During validation, the second half of the database was used. In this phase the input data is used, but the model

parameters were not changed. The objective is to examine the statistical consistency when comparing the known output to the estimated output (5:175-176). The validation data set entered in the models, and the results analyzed for accuracy. This validation should show that the model is an accurate predictor of effort in the environment of the calibration.

The third part of the research was a run of the independent data set through the models to examine the validity of the model outside its calibrated environment. The effort estimations were then analyzed for accuracy against the actual effort. The accuracy analysis should show that outside the environment of calibration, the models do not predict well, i.e. a model calibrated to a manned space environment should not give accurate estimates when used to estimate the effort necessary to develop a word processing application program.

To test the accuracy of the models, several statistical tests are used. The first tests are the coefficient of multiple determination (COMD or R^2) and the magnitude and mean magnitude of relative error. For the coefficient of multiple determination, Equation 3.1, E_{act} is the actual value from the database, E_{est} is the estimate from the model, and E_{mean}, Equation 3.2, is the mean of the estimated values. The COMD indicates the extent to which E_{act} and E_{est} are linearly related. The closer the value of

COMD is to 1.0, the better. (It is possible to get negative values for COMD if the error is large enough. The negative values appear when the difference between the actual effort and the estimate is extremely large.) A high value for COMD suggests that either a large percentage of variance is accounted for, or that the inclusion of additional independent variables in the model is not likely to improve the model estimating ability significantly. For the model to be considered calibrated, values above 0.90 are expected (5:148-176).

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (E_{act_i} - E_{est_i})^2}{\sum_{i=1}^{n} (E_{act_i} - E_{mean})^2} \qquad \text{Eq. 3.1}$$

$$E_{mean} = \frac{1}{n} * \sum_{i=1}^{n} E_{est_i} \qquad \text{Eq. 3.2}$$

The equation for magnitude of relative error (MRE) is Equation 3.3, and for mean magnitude of relative error (MMRE), Equation 3.4. A small value of MRE indicates that the model is preaicting accurately. The key parameter however, is MMRE. For the model to be acceptable, MMRE should be less than or equal to 0.25. The use of MRE and MMRE relieve the concerns of positive and negative errors

canceling each other and giving a false indication of model accuracy (5:148-176).

$$MRE = \left| \frac{E_{act} - E_{est}}{E_{act}} \right| \qquad \text{Eq. 3.3}$$

$$MMRE = \frac{1}{n} * \sum_{i=1}^{n} MRE_i \qquad \text{Eq. 3.4}$$

Errors using the MRE and MMRE tests can be of two types: underestimates, where $E_{est} < E_{act}$; and overestimates, where $E_{est} > E_{act}$. Both errors can have serious impacts on estimate interpretation. Large underestimates can cause projects to be understaffed and, as deadlines approach, project managers will be tempted to add new staff members, resulting in a phenomenon known as Brooks's law: "Adding manpower to a late software project makes it later" (4:25). Large overestimates can also be costly, staff members become less productive (Parkinson's law: "Work expands to fill the time available for its completion") or add "gold-plating" that is not required by the user (15:420).

The second set of statistical tests are the root mean square error (RMS), Equation 3.5, and the relative root mean square error (RRMS), Equation 3.6. The smaller the value of RMS the better is the estimation model. For RRMS, an acceptable model will give a value of RRMS < 0.25 (5:175).

$$RMS = \sqrt{\frac{1}{n}\sum_{n=1}^{n} (E_{act} - E_{est})^2} \qquad \text{Eq. 3.5}$$

$$RRMS = \frac{RMS}{\frac{1}{n}\sum_{n=1}^{n} E_{act}} \qquad \text{Eq. 3.6}$$

The third statistical test used is the prediction level test, Equation 3.7, where k is the number of projects in a set of n projects whose MRE is less than or equal to a percentage 1.

$$PRED(1) = \frac{k}{n} \qquad \text{Eq. 3.7}$$

For example, if PRED (0.25) = 0.83, then 83% of the predicted values fall within 25% of their actual values. To establish the model accuracy, 75% of the predictions must fall within 25% or the actual values, or PRED (0.25) >= 0.75 (5:173).

Summary

This chapter reviewed the data that was used for this research effort and the techniques to perform the

calibration, validation, and comparison of the models. The statistical techniques used were also presented.

Terminology

Source Lines of Code (SLOC) - all program instructions created by the project personnel and processed into machine code. It includes job control, format statements, etc., but does not include comment statements and unmodified utility software.

Man-month (MM) - generally consists of 152 man hours

IV. Analysis and Findings

Introduction

This chapter will present the analysis and finding of the research effort. First, an analysis of the databases will be presented, then the individual calibration, validation, and comparison analysis for each of the selected models.

Data

The data collection and analysis for this effort proved to be very frustrating. The original plan was to use a database from ESD (Electronic Systems Division of AF Systems Command). As the actual database was being analyzed for content, several key pieces of information were found to be missing or questionable. Several telephone conversations to ESD finally connected this researcher with Mr. Paul Funch of the Mitre Corporation. He pointed to a document he wrote which reviewed the database. His analysis of the database found it to be a very unreliable source of accurate software effort estimation model data. Several of the data points are incomplete; these points lack important pieces of multiplier information. Furthermore, several of the data points are for projects never completed. The data for these points, although either incomplete or estimated for completion, are included as actual data. For many of the

data points, the "actual" values entered are really not actuals. These values are "compromise" values agreed to by the company that collected the data, the Mitre people involved, and the ESD project office that oversaw the database collection effort (12:1-1 to 8-5; 11).

Because of the above problems with the ESD database, this researcher considers the accuracy of this database to be very suspect. This database can be used for example calibration and validation of estimation models, but for actual model development this database is not the best available.

For this research effort, this author had to turn to other sources for accurate data. One set of data was found at the Aerospace Corporation in Los Angeles (associated with Space Systems Division of the AF Systems Command). This database was found to be quite good; however, it arrived too late to be of use for this research project.

The data base that was used was the November 1990 version of a database collected by SSD/ACC. This updated database will eventually contain over 512 data points with a large amount of information for each point. The November 1990 version had enough data points, 150, that the methodology discussed in Chapter III could still be used. The actual data in this database or the Aerospace database cannot be published due to the proprietary nature of the data.

The SSD database was searched for at least 20 data points which could be used for the calibration and validation attempts. Twenty-eight data points were found that; had the same development environment (Military Ground Systems), had data for the actual development effort, had no reused code, and were similar sized projects. Having no reused code was a necessary requirement since the database does not include any information about the distribution of reused code, i.e. the amount of redesign, recode, etc., to determine the estimated source lines-of-code (SLOC) necessary for the model inputs. The selected project size ranged from 4.1K SLOC to 252K SLOC. Fourteen of the data points were used for the calibration effort and the other 14 for the validation effort. The selection of which 14 went to which effort was made by alternating the selection of the projects; the first went to the calibration effort, the second went to the validation effort, the third to calibration, etc.

For the comparison part of this research, 10 projects were found in the SSD database which fit all of the above criteria except for the development environment. The development environment selected was Unmanned Space Systems since data was available and this environment is different than Military Ground Systems.

REVIC

Since REVIC is a COCOMO derived estimation model, the technique described by Dr. Boehm (2:524-530) was used to perform the calibration. Dr. Boehm recommends at least 10 data points should be available for a coefficient and exponent calibration. Since 14 data points were available, the coefficient and exponent calibration was performed initially. However, since the number of data points was not large, this researcher decided to perform a coefficient only calibration also and compare the two calibrations. The semi-detached mode (Equation 4.1) of REVIC was used for the calibration and validation since the description of the projects selected from the SSD database for calibration and validation fit the description of Dr. Boehm's semi-detached mode, where MM is the output in man-months, kDSI is the source lines of code in thousands, and $\prod$ is the product of the costing parameters (2:74-80, 116-117).

$$MM=3.0x(kDSI)^{1.12}\prod \qquad \text{Eq. 4.1}$$

The embedded mode (Equation 4.2) was used in the comparison analysis for the coefficient only calibration since these data points match the description of Dr. Boehm's embedded mode description (2:74-80, 116-117).

$$MM=2.8x(kDSI)^{1.20}\prod \qquad \text{Eq. 4.2}$$

<u>Calibration</u>. The input data for the calibration effort is shown in Appendix B, Table B.1. The adjustment of these input values will give the calibrated coefficient and exponent or coefficient only values for this particular data set. For the coefficient and exponent calibration, the calibrated output values were 2.4531 and 1.2457 respectively. For the coefficient only calibration, the REVIC calibrated exponent of 1.20 was used. The calibrated coefficient was found to be 3.724.

These new coefficients and exponents were then put back into the estimation equations to look at prediction accuracies of the model for the data used for calibration (Appendix C, Table C.1 lists the estimates with the new calibration and percent of the actual effort.) Table 4.1 shows the results of the accuracy analysis.

Table 4.1 REVIC Calibration Accuracy Results

	Coefficient and Exponent	Coefficient Only
R^2	0.776	0.892
MMRE	0.3733	0.334
RMS	119.1416	82.641
RRMS	0.3192	0.221
PRED (0.25)	42%	57%

The interesting item of note here is that, for all the parameters, the coefficient only calibration appears to be more accurate than that of the coefficient and exponent.

This may be explained by the fact that the exponent calibration is very sensitive to small variations in project data (2:524-529). With a larger calibration data set the accuracy of the coefficient and exponent calibration may be better.

The other interesting item of note is the general accuracy of the calibrated model. Even against the calibration data, the model is not inherently accurate. R^2 should be greater than 0.90, MMRE and RRMS should be less than 0.25, RMS should be small (approaching 0), and PRED(0.25) should be greater than 75%. The coefficient only results approach acceptability as defined by Conte (5:150-176), but are nowhere near what should be expected of a model when tested against its calibration data.

Validation. The validation input data is shown in Appendix B, Table B.2. This data was used to try to validate the model as calibrated above. The results of the accuracy analysis are shown in Table 4.2.

Again, analysis of this table shows the coefficient calibration to be more accurate than the coefficient and exponent calibration. However, in this case both calibrations were able to predict four of the 14 validation projects to within 25% of their actuals. The differences in R^2, MMRE and RRMS show that the coefficient only calibration was more accurate, but none of the values are near what would be expected to say this model is validated to this environment (5:150-176).

Table 4.2 REVIC Validation Accuracy Results

	Coefficient and Exponent	Coefficient Only
R^2	0.1713	0.6583
MMRE	0.7811	0.6491
RMS	375.190	211.020
RRMS	0.8560	0.4815
PRED (0.25)	28.5%	28.5%

Comparison. The comparison input data is shown in Appendix B, Table B.3. This data was used to show how a model calibrated to one environment would predict in a completely different environment. The embedded mode was used for the coefficient only analysis with the new calibrated coefficient used. The results are shown in Table 4.3.

Table 4.3 REVIC Comparison Accuracy Results

	Coefficient and Exponent	Coefficient Only
R^2	0.9081	0.8381
MMRE	0.2201	0.1767
RMS	66.161	87.844
RRMS	0.2069	0.2748
PRED (0.25)	30%	70%

These results almost show this research effort to be futile, at least for the REVIC estimation model. The results show that both calibration efforts are fairly accurate with this set of data. Even though the PRED was

low, the other parameters are all very close to, if not, acceptable values. The R^2, MMRE, and RRMS show better results for the coefficient and exponent calibration, but the PRED and MMRE are much better for the coefficient only calibration. These results make this researcher question this model, using either the coefficient only or the coefficient and exponent calibration, as a valid effort estimation tool for any software manager. The model is too good at estimating outside the environment of calibration and not good at all inside the environment.

SASET

The research effort using the SASET estimation model was very frustrating. As this author reviewed the SASET model and User's Guide, the ability to calibrate the model was found to be virtually impossible. Since the mathematical equations published with the users guide are virtually impossible to understand, for the "average" user, and a calibration mode is not available as part of the computerized version of the model, this author could not figure out how to calibrate the model to a particular data set. The only way to perform a calibration was to go into the calibration file of the computerized model and change the actual values of several hundred different parameters. Without the knowledge of what each of these parameters actually does within the model, any changes would be pure guesswork. Again, the User's Guide was of no help. This

model has an unpublished saying that accompanies it, "There are no casual users of SASET." This saying seems very true, because an informal survey of normal users of effort estimation models revealed that they do not have the time, and sometimes not the mathematical abilities, to figure out the intricacies of this model.

Because of the above factors, a calibration of SASET was not accomplished. However, this research effort used SASET with its delivered calibration file and the 28 calibration and validation and 10 comparison data points were input to the model to test the model with its delivered calibration.

Calibration/Validation. Because of the proprietary nature, the complete data for each data point are not publishable with this effort. Appendix B includes the basics of the input parameters for the model. Table 4.4 shows the accuracy results for the calibration, validation, and comparison data sets. Appendix C, Table C.2 lists the estimation values and a comparison to the actual effort. As can be seen from the data, the existing calibration of SASET is very poor for this data set. The estimates were all greater than the actuals, with estimates from 2 to 16 times the actual values given as outputs from the model.

Table 4.4 SASET Accuracy Results

	Calibration/ Validation	Comparison
R^2	-0.7333	-0.3272
MMRE	5.9492	1.0985
RMS	1836.4	527.6
RRMS	4.5097	1.6503
PRED (0.25)	3.5%	0%

An expected value of R^2 greater than 0.90, MMRE and RRMS less than 0.25, RMS small (approaching 0) and PRED(0.25) greater than 75%, are considered acceptable to say a model is a good estimator (5:150-176). The negative values of R^2 are a result of the large differences between the actual effort and the estimate from the model (see Appendix C, Table C.1 for the data and Chapter 3 for the R^2 equation).

Comparison. The comparison data was analyzed with the SASET model to see if another environment was any better with the delivered calibration. As can be seen by the data in Table 4.4, the comparison data set also shows a very poor calibration for the data set. All of the estimates were greater than the actual efforts, nine of the ten data points were estimated between two and three times the actuals. This does at least show some consistently high estimation.

For the SASET model the computerized version is delivered with one specific calibration. For the layman

software effort estimator, this model has very questionable useability in its current form.

SEER

SEER was also found to be a problem for this research effort; however, this issue was not because of the usability (or unusability) of the model. The SEER model is calibratable, but only if the data set is properly annotated. The model has a parameter called "effective technology rating" which is used to calibrate the model to a particular environment or data set. To perform the evaluation of the effective technology parameter with a historical data set, the actual effort for the Full Scale Implementation phase (a SEER term) must be known. This phase does not include requirements analysis, or system integration and testing. The database that was used for this effort includes the necessary data, but not to the detail necessary to perform the calibration; i.e. the actual effort is known, but the effort during Full Scale Implementation is not. Again, the full database cannot be published with this effort due to its proprietary nature. (See Appendix B for the basic input data.)

Calibration/Validation. The 28 data points of the calibration and validation data set were ran through the model to test for model accuracy with this particular environment. Table 4.5 shows the results of this accuracy

analysis. Appendix C, Table C.3 lists the output results and the estimate as a percent of the actual effort values.

Table 4.5 SEER Accuracy Results

	Calibration/ Validation	Comparison
R^2	-1.0047	-0.2529
MMRE	3.5556	0.5586
RMS	1504.9	380.6
RRMS	3.6955	1.1905
PRED (0.25)	10.7%	20%

The estimates from the model ranged from 25% of the actual to 11 times the actual effort. Most of the estimates were in the range of 2-5 times the actual. The results shown in Table 4.5 again show the need to calibrate a model to a particular environment. R^2 is expected to be greater than 0.90, MMRE and RRMS are expected to be less than 0.25, RMS is expected to be small (approaching 0), and PRED(0.25) is expected to be greater than 75% for the model to be considered acceptable (5:150-176). R^2 is negative due to the large differences between the actual effort and the estimated effort (see Chapter 3 for the R^2 equation and Appendix C, Table C.3 for the data).

<u>Comparison</u>. The comparison data was also ran through the model. The results of the accuracy analysis are shown in Table 4.5. These results are some what better than those for the calibration and validation, but again this

model, as calibrated, should not be used in these environments. The estimates for this data set were all greater than the actual, ranging from very near the actual to three times the actual value.

The results of the accuracy analysis, especially the comparison data, lead this researcher to conclude that the SEER model may have some use if a proper calibration can be accomplished; but this will require a historical database that has the necessary effort information in each phase of the development life-cycle.

COSTMODL

The first review of COSTMODL revealed several differences between it, COCOMO, and REVIC. For this reason it was selected as a model to be evaluated. However, once the database issue was finally resolved, the only implementation of the model that was still valid (i.e. a non-Ada version) was that of the original COCOMO, adjusted to account for the Requirements Analysis and Operational Test and Evaluation phases. The procedure explained by Dr. Boehm (2:524-530) was used to perform the calibration.

Calibration. The input data for the calibration effort is listed in Appendix B, Table B.1. Since REVIC was analyzed for both the coefficient only and coefficient and exponent, COSTMODL was also. The derived coefficient only coefficient value was 4.255. The values for the coefficient and exponent analysis were 3.35 and 1.22 for

the coefficient and exponent respectively. These values were then used to replace the original coefficients and exponents in the model, and the model was analyzed for accuracy against the calibration data set. Table 4.6 shows these results. Appendix C, Table C.4 lists the estimates from the model and the estimate as a percent of the actual effort.

Table 4.6 COSTMODL Calibration Accuracy Results

	Coefficient and Exponent	Coefficient Only
R^2	0.5251	0.760
MMRE	0.4603	0.396
RMS	175.57	124.27
RRMS	0.4703	0.333
PRED (0.25)	29%	35.7%

Values of R^2 greater than 0.90, MMRE and RRMS less than 0.25, RMS small (approaching 0), and PRED(0.25) greater than 75% are expected for the model to be considered to be acceptable (5:150-176). These values are very similar to the accuracies shown with REVIC. This model is calibratable, but it still leaves a lot to be desired in the accuracy area. The coefficient only calibration appears to perform somewhat better against the calibration data set, but the performance increase is very small.

Validation. The validation data set (Appendix B, Table B.2) was ran and again analyzed for accuracy. The results are shown in Table 4.7.

Table 4.7 COSTMODL Validation Accuracy Results

	Coefficient and Exponent	Coefficient Only
R^2	0.1120	0.6353
MMRE	0.7863	0.5765
RMS	411.516	220.667
RRMS	0.9389	0.5035
PRED (0.25)	21.4%	21.4%

Again, the coefficient only calibration appears to be a better estimator of the actual effort. The results of this accuracy analysis show a questionable estimation model for the COSTMODL effort estimation, and the COCOMO baseline equations. These results are nowhere near what are necessary for a useable model within DoD.

Comparison. The comparison input data is listed in Appendix B, Table B.3. As with the other models, this data was used to see the effect of using a estimation model outside its calibrated environment. The accuracy analysis is shown in Table 4.8.

Analysis of this data shows that, according to the criteria of Chapter 3, this is a good calibration for this data set. This is not supposed to happen; a model should not work this well outside its calibrated environment.

This researcher does not understand why this model predicts well outside its environment of calibration.

Table 4.8 COSTMODL Comparison Accuracy Results

	Coefficient and Exponent	Coefficient Only
R^2	0.8661	0.8369
MMRE	0.2003	0.1751
RMS	79.454	87.94
RRMS	0.2485	0.2751
PRED (0.25)	70%	60%

The coefficient only analysis uses the embedded mode of Intermediate COCOMO, the same as with the REVIC comparison analysis.

Summary

This chapter presented the results and analysis of this research effort. The credibility of the database was reviewed and an attempt was made to calibrate, validate and compare each of the selected models. The results of this effort, for every model, show that the accuracies are not up to the level expected of an acceptable model.

V. Conclusions and Recommendations

Introduction

This chapter will summarize the research effort and offer some recommendations on where more research could and should be accomplished in this area.

Conclusions

This research proved to be very enlightening to this researcher. Based upon the background readings, this researcher believed that the existing marketed software effort estimation models were highly credible; however, this researcher found this not to be so based upon the research performed.

The two models that could be calibrated, REVIC and COSTMODL, could not predict the actuals against either the calibration data or validation data to any level of accuracy or consistency. Surprisingly, both of these models were relatively good at predicting the comparison data, data which was completely outside the environment of calibration. For the two models which were not calibrated, SASET and SEER, it was shown that calibration is necessary, but may not be sufficient to make either of these models usable. One interesting item that was found: During the initial attempts at calibrating REVIC and COSTMODL, one data point was used which had a significantly larger amount

of code than any of the others (over 700 KSLOC). This one data point was found to drive any attempt at calibration. The amount of code is one of the key terms used in the calibration technique for COCOMO and derivatives (2:524-530). This number is squared in several places as part of the calibration, and when one of the data points is much larger than the others, this squaring creates an extremely large number that can be magnitudes larger than those for the other data points. When these squared values are then summed, this one data point can drive the value of the sum. Therefore, this data point was removed from the calibration database.

REVIC proved to be a fairly easy model to learn and use. The calibration was not difficult and did produce an increased ability to estimate effort compared to the original calibration. However, the accuracy of this model is questionable based upon the results found in this research effort. This researcher found it interesting that the coefficient only calibration was actually more accurate than the coefficient and exponent calibration. This can probably be explained by the sensitivity of the exponent, but no way to test this is known by this researcher.

SASET proved to be the most difficult model to learn and use. The User's Guide is very unclear, and the model is not easy to learn and use just by running the computerized program. The calibration for this model will probably prove to be virtually impossible for any user

other than one of the model developers. This alone makes this model very difficult to use for any DoD acquisition program office since calibration is apparently needed. The model has many nice features and is very flexible in allowing risk analysis and trade-off analysis; but, if the model cannot be calibrated to the working environment, it probably cannot be used as an accurate predictor in a program office.

SEER was a fairly easy model to learn and use. The User's Guide is very well written and is easy to follow, once the template structure is learned. This model is relatively easy to calibrate if the historical data can be put into the necessary format. The inaccuracies found with the estimation analysis proved that SEER also needs to be calibrated to the operating environment. This should be done soon, since the AF will have a site license for this model beginning fiscal year 1992.

COSTMODL turned out to be very similar to REVIC. The model was very easy to learn, understand and use. Here the coefficient only calibration also seemed to work better than the coefficient and exponent calibration. This model proved to be calibratable, but again the poor accuracy results make it a questionable resource for any program manager.

Recommendations

One of the comments this researcher has heard throughout the graduate program was that software can now be "engineered;" hence, the term "software engineering." This researcher is not convinced this is true. In all the models evaluated, the two key factors that influenced the estimate were project size (SLOC) and the capabilities of the development team personnel. This researcher is not convinced that any effort estimation model that is so sensitive to the abilities of the development team can be applied across the board to any software development effort. These kinds of models might be useful to the actual development team for their own analysis and estimation, but for the user at the DoD system program office (SPO) level these models have little worth. The abilities of individual contractors are usually not known to any significant level, let alone the data for individual project teams. The user at the SPO level must make (sometimes educated) guesses about the information necessary to get an estimate from the model; this makes the estimates inherently inaccurate to use for any level of analysis. This brings up two important questions; 1) can the personnel data collected and used for a calibration really be validated as to the actual values, and 2) once the model is validated, what values is a user to use when estimating the effort for a project when the capabilities of the project team are quite open for interpretation, and

whatever values are used will cause major differences in the estimates calculated?

Several areas of research still need to be done in the area of software effort estimation. First is the area of estimating the SLOC (size of the development effort) and determining what exactly is a "line-of-code" in the various languages. Since the effort estimations are based upon a size estimate; this can incorporate inaccuracies into the results. Some effort is already underway in this area, Ikatura and Takayanagi equations, SPANS model by Tecolote Inc., Checkpoint estimating model, Bozoki Software Sizing Model, etc., but more is necessary (10).

Second is the area of cost drivers. Is there some way to tie the effort estimation to the capabilities of the needed system without including all the development team capability drivers? Is there some way to use other information to replace the personnel drivers, maybe by using the Software Engineering Institute (SEI) Process Model Maturity Level? For a particular SPO, the data being collected on contractor performance may be of use.

The third significant effort needs to be placed on the development of engineering practices to bring the software development into the realm of an engineering discipline. Once this is accomplished, there may be some new factors found which are drivers in the effort estimation techniques.

A fourth area is with the REVIC and COSTMODL models. Some effort should be undertaken to understand why these two models predicted the comparison data so well when the data was outside the environment of calibration.

A fifth area of research is in the area of the historical databases. More effort must be made to collect the necessary data to perform the calibration, validation and comparisons of the many effort estimation models. The effort must be placed in the collection of the data, understanding what needs to be collected, and the normalization of the data so it is usable in the various models.

Summary

This chapter summarized the research effort, made some conclusions based upon the effort, and made some recommendation on areas where further effort is necessary in the area of software effort estimation.

Appendix A: Other Related Important Documentation

1. Abdel-Hamid, T. and D. Madnick,. "Impact of Schedule Estimation on Software Project Behavior," IEEE Software, 3: 70-75 (Jul 1986).

2. Adelson, B. and E. Soloway. "The Role of Domain Experience in Software Design," IEEE Transactions on Software Engineering, SE-11: 1351-1360 (Nov 1985).

3. Albrecht, A. J. and J. Gaffney Jr. "Software Function, Source Lines of Code, and Development Effort Production: A Software Science Validation," IEEE Transactions on Software Engineering, SE-9: 639-648 (Nov 1983).

4. Amico, Vince. "A Software Risk Model," Simulation Series. 112-117. San Diego: Society of Computer Simulation, 1983.

5. Bailey, E. K. and others. "A Descriptive Evaluation of Automated Software Cost-Estimation Models," IDA Memorandum Report M-195, Institute for Defense Analysis, 1986. (AD-A178 400).

6. Belova, L. A. and V. V. Lipaev. "Cost Estimation of Complex Software Development for Management Systems," Automation & Remote Control, 49: 949-955 (20 Dec 1988).

7. Benbasat, I. and I. Vessey. "Programmer and Analyst Time/cost Estimation," MIS Quarterly, : 31-44 (June 1980).

8. Blalock, Crystal D. An Analysis of Schedule Determination in Software Program Development & Software Development Estimation Models. MS Thesis, AFIT/GCA/LSY 88S-2. School of Systems & Logistics, AFIT (AU), Wright-Patterson AFB OH, Sept 1988 (AD-A111204).

9. Boehm, Barry. "Software Engineering Economics," IEEE Transactions on Software Engineering, SE-10(1): 4-21 (1984).

10. Boehm, Barry. "Software Technology in the 1990s: Using an Evolutionary Paradigm," IEEE Computer, 30-37, (Nov 1983).

11. Boehm, Barry. "COCOMO: Answering the Most Frequent Questions," Proceedings of COCOMO Users Group, May 1985. Wang Institute, 1985.

12. Boehm, Barry. "Software Engineering," IEEE Transactions on Computing, C-25:1126-41 (Dec 1976).

13. Boehm, Barry. "Understanding and Controlling Software Costs," Information Processing, 86: 703-714, (1986).

14. Boehm, Barry. "The Hardware/Software Cost Ratio: Is it a Myth," IEEE Computer Journal, 16:78-80 (Mar 1983).

15. Bonan, Daniel. Decision Making Heuristics and Biases in Software Project Management: An Experimental Investigation, MS Thesis, Naval Postgraduate School, Monterey CA, Mar 1990, (AD-A226 678).

16. Callisen and others. "A Proposed Method for Estimating Software Cost and Requirements," Journal of Parametrics, 4: 33-40 (Dec 1984).

17. Chichinski, S. and G. S. Fowler. "Product Admin Through SABLE and NMAKE," AT&T Technical Journal, 67:59-70 (Jul/Aug 1988).

18. Clark, Gregory A. Software Cost Estimation Models -- Which one to use?, Report # 86-0545, Air Command and Staff College, Maxwell AFB AL, 1986, (AD-B102 721).

19. Cowderoy, A. J. C., and J. O. Jenkins. "State of the Art Survey for Software Cost-Estimation," Esprit P938 Report w/P5 la (issue 3). London: Imperial College Management School, 1986.

20. Cowderoy, A. J. C. and J. O. Jenkins. "Preparing for Knowledge-Based Cost-Estimation of Software Developments," Proceedings of the Joint International Symposium on Information Systems. Sydney: 1988.

21. Cowderoy, A. J. C. and J. O. Jenkins. "New Trends in Cost-Estimation," Proceedings of the 4th Annual Conference of the Centre for Software Reliability. Barking UK: Elsevier, 1986.

22. Cowderoy, A. J. C., and J. O. Jenkins. "Cost-Estimation by Analogy as a Good Management Practice," IEE Conference Publication n 290. 80-84. London England: IEE, 1988.

23. Dekker, G.J. A Software-Cost Database for the Development of Aerospace Software, NLR MP82044U, National Aerospace Laboratory (NLR), The Netherlands, 1983, (AD-B075 184).

24. Ditchenham, B. and N. R. Taylor. "Software Cost Models," ICL Tech Journal, 4:73-102 (May 1984).

25. Douville, A. and others. "ADA Impact on COCOMO, Workshop Report," Institute for Defense Analysis, (May 1985).

26. Dumas, R. I. "Final Report: Software Acquisition Resource Expenditure (SARE) Data Collection Model," Mitre Report MTR-9031. Mitre Corp. for Electronic Systems Division (AFSC), Hanscom AFB MA, 1983. (ESD-TR-83-214).

27. Eddins-Earles, M. "Cost Estimation Techniques for C3I System Software," RADC-TR-84-156. Rome Air Development Center, Griffis AFB NY, 1984.

28. Electronic Industries Association "DoD Computing Activities and Programs: Ten Year Market Forecast Issues, 1985-1995," Oct 1985.

29. Electronics Systems Division. A Review of Software Cost Estimation Methods. ESD-TR-76-271. Deputy for Command and Management Systems, Electronic Systems Division, Air Force Systems Command, USAF, Hanscom AFB MA, Aug 1976.

30. Ferens, Daniel V. "Software Parametric Cost Estimation: Wave of the Future," Engineering Costs and Production Economics, 14: 157-164 (Jul 1988).

31. Ferens, Daniel V. A Common Sense Approach to Software Cost Model Selection, 1986, (AD-B112 391).

32. Ferens, Daniel V. "Software Schedule Estimation: The Third Wave," Journal of Parametrics, X: 41-52 (Feb 1990).

33. Ferens, D. V. "Software Support Cost Models; Quo Vadis?," Journal of Parametrics, 4: 64-99 (Dec 1984).

34. Freiman, F. R. and R. E. Park. "PRICE Software Model ver. 3; An Overview," Proceedings, IEEE-PINY Workshop, Quantitative Software Models. 32-41. New York: IEEE Press 1979. [IEEE Cat No. TH 0067-9, Oct]

35. Gaffney, J. "Estimation of Software Code Based on Quantifiable Aspects of Function," Proceedings of the 6th ISPA Conference. 13-29. McLean VA: ISPA, 1984.

36. Gilb, Tom. "Software Cost Prediction vs. Software Cost Control," Journal of Parametrics, VII: 79-89 (Mar 1987).

37. Gulezian, Dr. R. Application of Statistical Methods to the Development of Naval Software Maintenance and Related Cost Estimation Models, RADC-TR-88-120, Rome Air Development Center, Griffiss AFB NY, May 1986, (AD-A201 840).

38. Hammonds, K. H. "Software: I.s a New Game," Business Week, 102-106, (4 Jun 1990).

39. Hanberg, A. C. and M. G. Miller. "Software Cost/ Size Estimation by Analogy," Report No. TR-5130-2-1, The Analytical Science Corporation. 30 June 1985.

40. Heemstra, F. J. and others. "Selection of Software Cost Estimation Packages," Research Report No. 36, University of Eindhoven, Eindhoven The Netherlands, 1989.

41. James, J. H. "Validation of Some Software Cost Estimation Models," Proceedings of 1984 Summer Computer Simulation Conference, vol. 2, Jul 23-24, 1984. 905-909. Boston: 1984.

42. Jensen, R. W. "A Comparison of the Jensen and COCOMO Schedule and Cost Estimation Models," Proceedings of the ISPA 6th Annual Conference, May 1984. 96-106. McLean VA: ISPA, 1984.

43. Kane, Patrick T. and others. "A Cost Model For Estimating the Costs of Developing Software in the ADA Programming Language," Proceedings of the Twenty-First Annual Hawaii International Conference on System Sciences. 782-790. New York: IEEE Press, 1988.

44. Kitchenham, B. A. and N. R. Taylor. "Software Project Development Cost Estimation," Journal of System Software, 5: (1985).

45. Kusters, R. J., and others. "Are Software Cost-Estimation Models Accurate?," Information and Software Technology, 32: 187-190 (3 Apr 1990).

46. Martin, Rick. "Evaluation of Current Software Costing Tools," ACM SIGSOFT Software Engineering Notes, 13: 49-51 (Jul 1988).

47. Masters, T. F. II. "An Overview of Software Cost Estimation at the NSA," Journal of Parametrics, 5: 72-84 (Mar 1985).

48. McGough, Keith. "Cost Estimation in Software Engineering," Journal of Parametrics, VII: 5-9 (Jun 1987).

49. McKenzie, M. and others. "A Dynamic Simulation Model of The Software Development Process," Proceedings of 1984 Summer Computer Simulation Conference, vol. 2, Jul 23-24, 1984. 899-904. Boston: 1984.

50. Najberg, Andrew C. "The TASC Software Cost and Requirements Estimator (TASCORE): A Top-Down, Structured Approach to the Software Cost Estimation Process," Proceedings of the National Aerospace & Electronics Conference 88. 686-691. New York: IEEE Press, 1988.

51. Putnam, L. H. and others. "A Tool for Planning Software Projects," Systems and Software, 147-154 (Jan 1984).

52. Putnam, L. and D. T. Putnam. "A Data Verification of the Software Fourth Power Tradeoff Law," Proceedings of the 6th ISPA Conference. 443-471. McLean VA: ISPA, 1984.

53. Rampton, J. "Unique Features of the JS-2 System for Software Development Estimation," Proceedings of the 6th ISPA Conference. 89-95. McLean VA: ISPA, 1984.

54. Reese, Richard M. and Jim Tamulevicz. "Software Sizing Methodologies," Journal of Parametrics, VII: 35-54 (Jun 1987).

55. Reifer, D. J. "A Poor Man's Guide to Estimating Software Costs," RCI-TR-012, Reifer Consultants, Inc., Torrance CA, 1984. (also RCI-TN-119)

56. Reifer, D. J. "ADA and its Impact on Cost Modeling," Tech Note RCI-TN-275, Reifer Consultants, Inc., Torrance CA, 7 Jul 1987.

57. Rome Air Development Center. Software Measurement Models: A DACS State of the Art Report (Draft). Data & Analysis Center for Software, Rome Air Development Center, Griffis AFB NY.

58. Rubens, H. A. "Productivity and Quality Strategies for Measurement," Fifth National Conference on Measuring Data processing Quality and Productivity. Orlando FL: Quality Assurance Institute, 1987.

59. Rubin, Howard A. and others. "A Comparison of Cost Estimation Tools (A Panel Session)," International Conference on Software Engineering, 8th Publication. 174-180. New York: IEEE Press, 1985.

60. Rubin, H. A. "The Art and Science of Software Estimation: Fifth Generation Estimaters," Proceedings of the 7th ISPA Conference, Orlando, Fl, May 7-9, 1985. 56-72. McLean VA: International Society of Parametric Analysts, 1985.

61. Ryback, W.H. "Strengths & Limitations of Some Software Cost Estimation Methods," TOR-0083(3902-03)-3, Aerospace Corporation, Jul 1983.

62. Setzer, Robert. "Spacecraft Software Cost Estimation: Striving for Excellence Through Parametric Models (A Review), SAE Special Publication SP-636. 47-55. Warrendale PA: SAE, 1985.

63. Shen, V. Y, and others. "A Comparison of a Few Effort Estimation Models," Proceedings of the 6th ISPA Conference. 30-47. McLean VA: ISPA, 1984.

64. Sierevelt, H. "Observations on Software Models," Journal of Parametrics, 6: (Dec 1986).

65. Stanley, Margaret. "Software Cost Estimating," Journal of Parametrics, 4: 52-85 (Sept 1984).

66. Stewart, Nick. "Software Error Costs," Quality Progress, 21: 48-49 (Nov 1988).

67. Theabaut, S. M. "Model Evaluation in Software Metrics Research," Computer Science and Statistics: Proceedings of the 15th Symposium on the Interface (Houston, TX, Mar 83). 277-285. 1983.

68. Thebaut. "An Analytical Resource Model for Large-Scale Software Development," Information & Processing Management, 20: 293-315 (1984).

69. Tutorial. "Software Cost Estimation and Life Cycle Control," IEEE Computer Society Press, 1980.

70. Van Patten King, C. and others. "SoftCost-R User's Manual," Report No. RCI-TN-118 (vol 6.0), Reifer Consultants Inc., Torrance CA, 1987.

71. Waldrop, M. M. "Congress Finds Bugs in the Software," Science, 246:753 (10 Nov 1989).

72. Walston, C. E. and C. P. Felix. "A Method of Programming Measurement and Estimation," IBM System Journal, 16(1): 54-73 (1977).

73. Wheaton, M. J. "Software Sizing Task Final Report," ATM-84(45-2303)-1, Aerospace Corporation, 1984.

74. Wheaton, M. J. "Functional Software Sizing Methodology," ISPA Journal of Parametrics, 6(1): 17 (1986).

75. Whetstone, M. J. Developing Software Size Estimating Relationships Based on Functional Descriptions of the Software. MS Thesis GSM/LSY, School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, Sept 1986.

76. Zultner, R. "The Deming Approach to Software Quality Engineering," Quality Progress, (Nov 1988).

Appendix B: Input Data

Table B.1 Calibration Data

Project Number	Lines of Code (thousands)	Actual Effort (man-months)	REVIC Product Factor	COSTMODL Product Factor
18	70.143	658	1.299	0.855
24	18	238	1.442	0.95
28	112.917	887	1.119	0.811
37	11.829	136	0.933	0.55
41	9.5	13	0.617	0.474
48	45.068	405	1.183	1.183
51	37.836	193	1.432	1.203
53	100.505	540	0.985	0.827
55	96.059	589	0.951	0.951
57	137.804	697	0.827	0.827
59	12.862	229	0.993	0.835
61	36.138	134	0.856	0.719
63	61.752	337	0.856	0.719
71	13	170	2.491	1.24

Table B.2 Validation Data

Project Number	Lines of Code (thousands)	Actual Effort (man-months)	REVIC Product Factor	COSTMODL Product Factor
09	128.2	545	0.908	0.909
23	5.2	42	1.586	1.045
26	4.17	227	1.586	1.045
30	41	160	2.061	1.358
39	17	19	0.891	0.587
46	20	103	0.646	0.588
50	71.676	473	1.047	1.407
52	177.06	840	1.030	0.866
54	148.29	688	1.132	0.951
56	252.87	1194	0.721	0.606
58	88.679	687	1.112	0.934
60	5.846	172	0.855	0.719
62	56.333	535	1.178	0.99
65	144	656	0.791	0.869

Table B.3 Comparison Data

Project Number	Lines of Code (thousands)	Actual Effort (man-months)	REVIC Product Factor	COSTMODL Product Factor
288	11.7	80	0.972	0.972
289	116.8	912	0.993	0.720
290	14	115	0.829	0.601
291	56.2	523	1.220	0.884
292	48.3	478	1.130	1.130
293	50.3	432	0.972	0.972
294	69.54	296	0.884	0.884
295	22.9	164	0.884	0.884
296	16.3	140	1.494	1.494
267	6.8	57	0.884	0.884

Appendix C: Model Estimates after Calibration

Table C.1 REVIC Estimates

Project Number	Coefficient & Exponent	% of actual	Coefficient only	% of actual
09	959.65	176.1	776.1	142.4
18	635.15	96.5	565.2	85.9
23	30.51	72.6	37.43	89.1
24	129.53	54.4	136.75	57.5
26	23.16	105.2	29.23	132.9
28	990.1	111.62	829.81	93.6
30	523.87	327.4	491.37	307.1
37	49.68	36.5	55.29	40.7
39	75.35	396.6	79.25	417.1
41	25.0	192.3	28.60	220.0
46	66.94	65.0	68.93	66.9
48	333.38	82.3	313.6	77.4
50	534.97	113.1	466.63	98.7
51	324.54	168.2	312.07	161.7
52	1629.87	194.0	1263.99	150.5
53	753.86	139.6	641.13	118.7
54	1435.20	208.6	1138.97	165.5
55	687.95	116.8	588.41	99.9
56	1781.23	149.2	1318.82	110.5
57	937.81	134.6	766.54	110
58	741.39	107.9	629.03	91.6
59	58.69	25.6	64.63	28.2
60	19.04	11.1	23.01	13.4
61	183.21	136.7	177.2	132.2
62	445.42	83.3	400.88	74.9
63	357.12	106.0	322.89	95.8

Table C.1 REVIC Estimates (continued)

Project Number	Coefficient & Exponent	% of actual	Coefficient only	% of actual
65	966.71	147.4	770.12	117.4
71	149.19	87.8	164.08	96.5
288	51.53	64.4	69.26	86.6
289	934.16	102.4	1119.16	122.7
290	55.0	47.8	52.76	63.7
291	459.94	87.9	571.55	109.3
292	352.52	73.7	441.39	92.3
293	319.0	73.8	398.62	92.3
294	434.92	145.9	534.74	180.7
295	108.5	66.2	141.02	86.0
296	119.88	85.6	158.48	113.2
297	23.78	41.7	32.85	57.6

Table C.2 SASET Estimates

Project Number	Model Estimate	% of actual
09	2490	456.9
18	1358	206.4
23	112	266.7
24	383	106.9
26	90	409.1
28	2187	246.6
30	920	575.0
37	229	168.4
39	305	1605.3
41	169	1300.0
46	328	318.4
48	906	223.7
50	1424	301.1
51	782	405.2
52	3268	389.0
53	2030	375.9
54	2995	435.3
55	1887	320.4
56	4667	390.9
57	2707	388.4
58	1791	260.7
59	241	105.2
60	116	67.4
61	721	538.1
62	1137	212.5
63	1233	327.1

Table C.2 SASET Estimates (continued)

Project Number	Model Estimate	% of actual
65	2909	443.4
71	282	165.9
288	232	290.0
289	2313	253.6
290	271	235.7
291	1113	212.8
292	949	198.5
293	988	228.7
294	1249	422.0
295	412	251.2
296	324	231.4
297	122	214.0

Table C.3 SEER Estimates

Project Number	Model Estimate	% of actual
09	1768	324.4
18	1032	156.8
23	73	173.8
24	215	90.3
26	40	181.8
28	2682	302.4
30	663	414.4
37	100	73.5
39	209	1100.0
41	54	415.4
46	126	122.3
48	724	178.8
50	891	188.4
51	662	343.0
52	3343	398.0
53	1632	302.2
54	688	505.4
55	1231	209.0
56	3644	305.2
57	1798	258.0
58	1746	254.1
59	198	86.5
60	44	25.6
61	412	307.5
62	914	170.8
63	770	204.2

Table C.3 SEER Estimates (continued)

Project Number	Model Estimate	% of actual
65	1950	297.3
71	239	140.6
288	119	148.8
289	1991	218.3
290	115	115.7
291	523	195.0
292	773	161.7
293	714	165.3
294	944	318.9
295	251	153.0
296	235	203.6
297	58	101.8

Table C.4 COSTMODL Estimates

Project Number	Coefficient & Exponent	% of actual	Coefficient only	% of actual
09	1135.7	208.4	887.8	162.9
18	518.6	78.8	424.9	64.6
23	26.2	62.3	28.2	67.1
24	109.1	45.9	102.9	43.2
26	20.0	90.8	22.0	100.0
28	880.7	99.3	687.0	77.5
30	422.2	263.9	260.8	231.2
37	37.8	27.8	37.2	27.4
39	62.4	328.2	59.7	314.0
41	24.9	191.6	25.1	193.1
46	76.2	73.9	71.7	69.6
48	417.7	103.1	358.2	88.5
50	864.8	182.8	716.5	151.5
51	342.9	177.7	299.5	155.2
52	1604.2	191.0	1214.3	144.6
53	778.8	144.2	614.9	113.9
54	1419.0	206.3	1093.3	158.9
55	847.4	143.9	672.2	114.1
56	1734.0	145.2	1266.5	106.1
57	1145.9	164.4	875.7	125.6
58	744.3	108.3	603.7	87.9
59	63.9	27.8	62.1	27.1
60	20.8	12.1	22.1	12.9
61	193.8	144.6	170.0	127.9
62	453.6	84.8	384.9	72.0
63	373.2	110.7	309.8	91.9

Table C.4 COSTMODL Estimates (continued)

Project Number	Coefficient & Exponent	% of actual	Coefficient only	% of actual
65	1251.0	190.7	966.7	147.4
71	95.7	56.3	93.3	54.9
288	65.5	81.8	79.1	98.9
289	802.9	88.0	927.2	101.7
290	50.4	43.8	60.7	52.8
291	403.8	77.2	473.2	90.5
292	429.1	89.8	504.3	105.5
293	387.8	89.8	455.5	105.4
294	523.6	176.9	611.0	206.4
295	135.1	82.3	161.1	98.2
296	150.8	107.7	181.1	129.3
297	29.3	51.4	35.8	62.9

Bibliography

1. Abdel-Hamid, Tarek K., and others. "On the Portability of Quantitative Software Estimation Models," Information & Management, 13: 1-10 (Aug 1987).

2. Boehm, Barry. Software Engineering Economics, Englewook Cliffs NJ: Prentice Hall, 1981.

3. Boehm, Barry and Philip N. Papaccio. "Understanding and Controlling Software Costs," IEEE Transactions on Software Engineering, 14: 1462-1477 (Oct 1988).

4. Brooks, Frederick P. Jr. The Mythical Man-Month, Menlo Park CA: Addison-Wesley, 1982.

5. Conte, Samuel D., and others. Software Engineering Method and Metrics. Menlo Park CA: Benjamin/Cummings, 1986.

6. COSTMODL User's Guide, NASA Johnson Space Center, Houston TX, version 5.2, Jan 1991.

7. Cover, Donna K. "Issues Affecting the Reliability of Software-Cost Estimates (Trying to Make Chicken Salad Out of Chicken Feathers)," Proceedings of the Annual Reliability and Maintainability Symposium. 195-201. New York: IEEE Press, 1988.

8. Cuelenaere, A.M.E., and others. "Calibrating a Software Cost Esitmation Model: Why & How," Information and Software Technology, 29: 558-567 (10 Dec 1987).

9. Editorial, Aviation Week & Space Technology, 129: 15 (17 Oct 1988).

10. Ferens, Daniel V. Class notes from IMGT 676, Software Cost Estimation. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, Mar 1991.

11. Funch, Paul G. Telephone interview. Mitre Corporation, Hanscom AFB MA, 1 Apr 1991.

12. Funch, Paul G. "Software Cost Data Base," MTR 10329 Rev 1, Mitre Corporation, May 1989.

13. Jensen, R.W. and S. Lucas. "Sensitivity Analysis of the Jensen Software Model," Proceedings of the 5th ISPA Conference. McLean VA: ISPA, 1983.

14. Jensen, R.W. "An Improved Macrolevel Software Development Resource Estimation Model," Proceedings of the 5th ISPA Conference. McLean VA: ISPA, 1983.

15. Kemerer, Chris F. "An Empirical Validation of Software Cost Estimation Models," Communications of the ACM, 30: 416-429 (May 1987).

16. Kile, Raymond L. REVIC Software Cost Estimating Model User's Manual, version 9.0, Feb 1991.

17. Lehder, Wilfred E. and others. "Software Estimation Technology," AT&T Technical Journal, 67: 59-70 (Jul/Aug 1988).

18. McCallam, Dennis H. and Michael J. Campbell. "Software Cost During System Procurement," Proceedings of the National Aerospace and Electronics Conference. 694-699. New York: IEEE Press, 1986.

19. Miyazaki, Yukio, and Kuniaki Mori. "COCOMO Evaluation and Tailoring," Proceedings of the International Conference on Software Engineering 8th Publication. 292-299. New York: IEEE Press, 1985.

20. Morrocoo, John D. "War Will Reshape Doctrine, But Lessons are Limited," Aviation Week & Space Technology, 40-43 (22 Apr 1991).

21. Reifer, Donald J. "SoftCost-R: User Experiences and Lessons Learned at the Age of One," The Journal of Systems and Software, 7: 279-286 (Dec 1987).

22. Schlender, Brenton R. "How to Break the Software Logjam," Fortune, 120: 100-101+ (25 Sep 1989).

23. SEER User's Manual, Galorath Associates Inc., Marina Del Rey CA, Aug 1989.

24. Silver, Dr. Aaron, and others. SASET Users Guide, Publication R-0420-88-2, Naval Center for Cost Analysis, Department of the Navy, Washington DC, Feb 1990.

Vita

Captain Gerald L. Ourada was born on 11 September 1959 in Boise, Idaho. He graduated from Capital High School in 1977, and graduated from the University of Idaho with a Bachelor of Science in Electrical Engineering in May 1982. His activity duty military career started in October 1981, when he was recruited into the Air Force's College Senior Engineering Program. He attended Officer Training School and received his commission in August 1982. His first duty station was to the 6595 Missile Test Group, Vandenberg AFB, CA, where he served testing the MX (now Peacekeeper) missile. His duties included; processing the solid stages in preparation for flight, readying the Minuteman launch system at Vandenberg for integration with the MX, and basing system advisor to the MX mission controller. His next duty assignment was at Space Systems Division, Los Angeles AFB in March 1986, where he was assigned to the Office of Plans and Advanced Programs. He was the lead space mission requirements analyst and spent many days TDY to Peterson AFB to work requirements issues with AFSPACECOM. In May 1989 he changed offices at SSD and became chief of the requirements section for the Defense Dissemination System. In May 1990 he entered the School of Systems and Logistics, Air Force Institute of Technology for a new program, Software Systems Management.

Permanent Address: 1122 Ourada Ranch Rd.
Boise, Idaho 83703

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Master's Thesis

4. TITLE AND SUBTITLE
SOFTWARE COST ESTIMATION MODELS: A CALIBRATION, VALIDATION, AND COMPARISON

5. FUNDING NUMBERS

6. AUTHOR(S)
Gerald L. Ourada, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSS/LSY/91D-11

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This study was a calibration, validation and comparison of four software effort estimation models. The four models evaluated were REVIC, SASET, SEER, and COSTMODL. A historical database was obtained from Space Systems Division, at Los Angeles AFB, and used as the input data. Two software environments were selected, one used to calibrate and validate the models, and the other to show the performance of the models outside their environment of calibration. REVIC and COSTMODL are COCOMO derivatives and were calibrated using Dr. Boehm's procedure. SASET and SEER were found to be uncalibratable for this effort. Accuracy of all the models was significantly low; none of the models performed as expected. REVIC and COSTMODL actually performed better against the comparison data than the data from the calibration. SASET and SEER were very inconsistent across both environments.

14. SUBJECT TERMS
Software Cost Models, Software Cost Estimates, Software Cost Analysis, Software Life Cycle Costs

15. NUMBER OF PAGES
78

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT Control Number AFIT/GSS/LSY/91D-11

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

Man Years ________________ $________________________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

________________________________ ________________________________
Name and Grade Organization

________________________________ ________________________________
Position or Title Address